

3D Spieleprogrammierung mit JAVA

Wichtige Befehle & Vorgehensweisen

Modelle

Modelle laden	<code>Spatial s = assetManager.loadModel(<PATH>);</code>
Modelle skalieren	<code>s.scale(x,y,z);</code>
Modelle der rootNode hinzufügen/entfernen	<code>rootNode.attachChild(s);</code> <code>rootNode.detachChild(s);</code>
Kollisionsform erzeugen	<code>CollisionShape shape = new</code> <code>BoxCollisionShape(new Vector3f (x, y, z));</code>
RigidBodyControl erzeugen	<code>control = new RigidBodyControl(shape, 0);</code>



Im SceneComposer navigieren

Zoomen	Scrollen
Kamera drehen	linke Maustaste
In der Welt verschieben	rechte Maustaste
Geometry eines Modells anzeigen	farbiger Würfel
Modelle hinzufügen	rechte Maustaste auf das Modell, dann „Link in SceneComposer“
Modelle bearbeiten	in den Properties (rechts)



Audio erstellen

Audio laden	<code>new AudioNode(assetManager, <PATH>, false);</code> <i>Hinweis: Letzter Parameter ermöglicht Audio-Buffering.</i>
Audio wiederholen lassen	<code>myAudioNode.setLooping(true);</code>
Räumliches Audio deaktivieren	<code>myAudioNode.setPositional(false);</code> <i>Hinweis: Räuml. funktioniert nur bei Stereo-Sounds.</i>
Lautstärke festlegen	<code>myAudioNode.setVolume(float f);</code>
Audio hinzufügen	<code>rootNode.attachChild(myAudioNode);</code>
Audio abspielen / starten	<code>myAudioNode.play();</code>
Audio anhalten	<code>myAudioNode.stop();</code>



Tutorium – Aufgaben

Aufgabe 1 [Modelle & Kollisionen erzeugen]

In dieser Aufgabe soll der Wald für das Spiel manuell im Code erzeugt werden. Verwende im Folgenden die **rootNode** um Elemente zum Spiel hinzuzufügen.

1. Gehe dazu in die *Forest* Klasse und erstelle ein neues *Spatial* für einen Baum. Der entsprechende Pfad für das Modell ist <<Models/Tree/Tree.mesh.j3o>>
2. Anschließend muss der Baum richtig skaliert werden. In y-Richtung soll der Baum 4-fach vergrößert werden, in x und z – Richtung 2-fach.
3. Damit man nicht durch die Bäume gehen kann, soll nun eine Kollisionsform um den Baum erstellt werden. Erstelle dazu eine *BoxCollisionShape* mit einer Breite von 0.3 (Quadrat) und einer Höhe von 10.
4. Erstelle etwas weiter unten in der for-Schleife ein *RigidBodyControl* *treeNode* und übergib die in 3. erstellte *shape* dem *RigidBodyControl*. Dieser überwacht ob Modelle kollidieren.
5. Füge die erstellte *RigidBodyControl* den einzelnen Bäumen hinzu. Dies funktioniert über den Befehl *addControl()* auf dem entsprechenden *Spatial*.
6. Füge die Bäume nun der *rootNode* hinzu

Du kannst nun das Spiel starten und nach kurzer Wartezeit (*die gesamten Modell müssen geladen werden...*) überprüfen ob alles richtig war.

Aufgabe 2 [Modelle dem Scenefile hinzufügen]

Öffne die *world.j3o* Datei in *ProjectAssets/Models/Scenes* im *SceneComposer* (durch Doppelklick). Füge nun beliebige Modelle aus dem *Models*-Ordner unter hinzu. Klicke dafür auf ein Modell mit der rechten Maustaste und wähle „Link in SceneComposer“ aus. Experimentiere nun mit einigen Modellen in den Properties herum (*scale*, *rotate*, *LocalTranslation*) und mache dich mit dem *SceneComposer* vertraut.

Tipp: Auf der ersten Seite findest du Hinweise zum Navigieren.

Aufgabe 3 [Sounds erstellen]

Zu guter Letzt sollen weitere Sounds im Game erzeugt werden, da diese für die entsprechende Atmosphäre sorgen. Verwende im Folgenden die **rootNode** um Elemente zum Spiel hinzuzufügen.

1. Um direkt eine gruslige Atmosphäre zu erhalten, soll bei Spielstart ein Gewitter abgespielt werden. Deine Aufgabe ist es die entsprechende **WAV-Datei zu laden und abzuspielen**.
 - 1.1) Gehe dazu in die von uns erstellte Klasse *AudioManager*. Hier ist eine Sammlung aller Audio-Dateien die im Spiel verwendet werden.
 - 1.2) In der Methode *initAudio* werden alle Sound-Files initialisiert. Erzeuge nun ein Sound-Objekt *audio_nature*. Orientiere dich dabei an den Beispielen in der Klasse und nimm die folgenden Informationen zur Hilfe:

Pfad: „Sounds/soundFX/thunder2.wav“
Lautstärke: 0.06


Räumlicher Sound: Nein
Wiederholendes Abspielen: Ja

- 1.3) Füge die *AudioNode* zum Spiel hinzu und starte sie direkt in der *initAudio*-Methode, da diese ab Spielbeginn abgespielt werden soll.
2. Analog wie unter 1. sollen nun die Fußstapfen im Wald erzeugt werden.

2.1) Pfad:
„Sounds/sound_fx_foodsteps1.wav“
Lautstärke: 0.4

Räumlicher Sound: Nein
Wiederholendes Abspielen: Ja

- 2.2) Füge auch hier nochmals die *AudioNode* zum Spiel hinzu.
 - 2.3) Nun sollen die Fußstapfen nicht ständig abgespielt werden, sondern nur dann, wenn sich der Spieler nach vorne bewegt. Gehe hierzu in die *main*-Methode und navigiere zum *AnalogListener*. Füge in die entsprechende *if*-Bedingung das Abspielen des Sound-Effektes ein. (Die Logik haben wir für dich übernommen...)
Hinweis: Die AudioManager – Klasse wird in der main-Methode erzeugt und steht einem dort mit allen Sounds als Instanzvariablen zur Verfügung...

Probiere alles über deine Kopfhörer aus 😊. 

(Zusatzaufgabe) Aufgabe 4 [Bug Fixes]

Glückwunsch! Du hast alle Aufgaben geschafft. Leider haben sich in der *main*-Methode im Spiel einige kleiner Fehler eingeschlichen. Mal sehen ob du alle beheben kannst...

1. Wie dir vielleicht schon aufgefallen ist, erscheint leider keine Benutzeraufforderung, wenn man vor einem Buch steht. Versuche in der *updateItemCollision*-Methode eine Logik zu implementieren, welche bei einem Abstand < 3 Meter eine entsprechende Meldung ausgibt.
Hinweis: Die Distanz wird bereits in der findNextBook Methode berechnet. Außerdem könnte dir die Methode showHUD helfen... (Head-up-display)
2. Leider sieht das Feuer noch etwas eckig aus. Bekommst du es in der *makeFire*-Methode hin auf den *ParticleEmitter* eine schöne Textur zu legen? Versuch's mal mit *setTexture* und dem Pfad „Effects/Explosion/flame.png“.
3. Leider kann der Spieler noch nicht springen. Rufe im passenden Listener an der richtigen Stelle eine *jump*-Methode auf dem Spieler auf.
Hinweis: Analog-oder ActionListener...